Claims

[c1] 1. A computer-based method and article of manufacture for graphical development of fully executable workflow applications, wherein said fully executable workflow applications are in form of computer-executable program code for loading in computer memory code segments and, after said computer-executable program code being loaded and, after said workflow applications being run by invoking said program code loaded in code segments, said running program code facilitates construction of necessary objects and threads in a way reflecting desired workflow configuration interactively described during applications" graphical development, wherein said necessary objects and threads facilitate processing of workflow orders in a way described during applications" graphical development.

[c2] 2. A computer-based method and article of manufacture as per claim 1, further including and implementing a method of defining of a workflow-process comprising following steps:

(A) Defining workflow-activities matrix, wherein said workflow-activities are individual items of work com-

prised by potential workflow-process where all participating elements of work included in said item are potentially executed as a single transaction within a said workflow-activity of said potential workflow-process, wherein said defining comprises following steps:

(i) Arranging workflow-activities comprised by the workflow-process being defined according to required sequence of execution of said workflow-activities within said workflow-process and according to possibilities for concurrent execution of some of said workflow-activities within same request processing within said workflow-process;

(ii) Defining a constant, named PROCESS_STEPS in preferred embodiment, representing number of sequential steps of execution of this workflow-process, wherein on each said step either one activity is executed either two or more activities might be executed concurrently within same request processing;

(iii) Defining a constant, named MAX_STEP_DEPTH in preferred embodiment, representing maximum number of workflow-activities that might be executed concurrently within same request processing

on each step of workflow-process being defined;

(iv) Declaring and initializing a two-dimensional array of Boolean elements being a workflow-activities matrix, with dimensions MAX_STEP_DEPTH by PROCESS_STEPS, wherein an array element with value TRUE represents existence of a workflow-activity on position in said workflow-activities-matrix with coordinates equal to said array element indexes and wherein an array element with value FALSE represents non-existence of a workflow-activity on position in said workflow-activities matrix with coordinates equal to said array element indexes.

(B) Defining main flow-graph, wherein defined at step (A) workflow-activities further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within potential workflow-process, wherein each said control-connector potentially signals successful execution of its sourcing workflow-activity with execution result having value Y, wherein two or more control-connectors sourcing from one said node might initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors

having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node, wherein said defining comprises following steps:

(i) Defining a constant, named MAX_CONNECT_OUT in preferred embodiment, representing maximum number of main control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX_CONNECT_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS_STEPS-1), wherein values of each pair of elements of said array represent indexes in initialized at step (A (iv)) workflow-activities array andthereby describe position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_CONNECT_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX_CONNECT_OUT

control-connectors having common source workflow-activity with potential execution result having value Y;

(iii) Optionally combining all declared and initialized at step (ii) arrays into a single four-dimensional array with dimensions MAX_STEP_DEPTH by PROCESS_STEPS-1 by MAX_CONNECT_OUT by 2.

(C) Optionally defining one or more alternative control-connectors, wherein each said alternative control-connector potentially signals non-successful execution of its sourcing workflow-activity with execution result having value N being different from non-successful execution result having value T requiring termination of processing of a particular workflow request, wherein two or more alternative control-connectors sourcing from one workflow-activity might initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established at step (B) a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control-connector, wherein said defining comprises following steps:

(i) Defining a constant, named

MAX_ALTCONNECT_OUT in preferred embodiment, representing maximum number of alternative control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX_ALTCONNECT_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to PROCESS_STEPS, wherein values of each pair of elements of said array represent indexes in initialized at step (A (iv)) workflow-activities array andthereby describe the position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_ALTCONNECT_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX_ALTCONNECT_OUT alternative control-connectors having common source workflow-activity with potential execution result having value N;

(iii) Optionally combining all declared and initialized at step (ii) arrays into a single four-di-

mensional array with dimensions MAX_STEP_DEPTH by PROCESS_STEPS by MAX_ALTCONNECT_OUT by 2;

(D) Defining workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of workflow-process being defined, for potential plugging for execution as part of a single transaction within said workflow-activity of potential said workflow-process, wherein said defining comprises following steps:

(i) Declaring a two-dimensional array of elements of type UUID (Universal Unique Identifier, interchangeable with the term GUID, Globally Unique Identifier) with dimensions identical to dimensions of declared at step (A (iv)) array representing workflow-activities matrix;

(ii) Initializing elements of array declared in step (i) in following manner: for every element of said array representing workflow-activities matrix with value TRUE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value representing a software component to be associated with repre-

sented workflow-activity, and for every element of said array representing workflow-activities matrix with value FALSE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value indicating a dummy UUID.

[c3] 3. A computer-based method and article of manufacture as per claim 2, further including generation of source code and compiling and linking instructions, wherein said source code and compiling and linking instructions are necessary and sufficient to build fully executable workflow application with third party produced and available on the market compilers and linkers, wherein after said source code is compiled and linked, and after produced executable program code is loaded and executed, said executable program code facilitates construction of hierarchical tree of objects reflecting specified, as per claim 2, workflow process description.

[c4] 4. A computer-based method and article of manufacture producing configuration of class objects and threads with capacity for concurrent processing of multitude of transactional workflow requests of identical type, wherein said configuration of class objects and threads represents a workflow processing configuration corre-

sponding to a particular workflow process description, wherein said configuration of class objects and threads represents control flows between workflow-activities, including optional launch of parallel control flows within same workflow request processing and parallel control flows synchronization, wherein said capacity for concurrent processing includes capacity of every one of workflow activities participating in said workflow process for concurrent processing of multitude of transactional workflow requests, wherein said configuration of class objects and threads has ability for adaptation to changing working conditions related to delays in distributed environment and fluctuations in received workload, wherein said adaptation has form of run-time self-adjustment of objects and threads configuration.

[c5] 5. A computer-based method and article of manufacture as per claim 4, further including construction of hierarchical tree of class objects with capacity to represent variety of workflow configurations, wherein said class objects are computer memory instances of classes or structures, wherein any level of said hierarchical tree contains one or multiple collections, each said collection containing predefined or dynamically defined number of one or multiple objects, wherein objects belonging to any said collection of same hierarchy level are computer

memory instances of classes or structures of identical type, wherein first level of said hierarchical tree consists of a single collection and every next level of hierarchy contains one or multiple collections and possibly individual objects, wherein said collections and said individual objects being accessible via references controlled by objects belonging to collections from previous level, wherein said hierarchical tree of class objects contains following levels and siblings:

(A) First level, a steps-collection, wherein said collection objects represent steps of said workflow-process, each said object being parent of a Second level collection;

(B) Second level, an activities-collection per step of said workflow-process, wherein said collection objects represent workflow-activities that might be executed concurrently with all other workflow-activities represented by objects belonging to same collection, wherein each said object is parent of Third level siblings, said siblings being of 3 distinct categories: a processor collection, a workload-balancer object, and optionally a workflow-synchronization object;

(C) Third level's sibling One, a processors-collection per workflow-activity, wherein an object of said collection represents an individual workload-processing squad having a capacity to process a portion of en-

tire workload specific for said workflow-activity, each said object being parent of a Fourth level collection;
(D) Third level's sibling Two, a workload-dispatching object per workflow-activity;
(E) Third level's optional sibling Three, a workflow-synchronizing object per workflow-activity if required by flow-graph of said workflow-process, wherein said object contains data defining required synchronization scheme;
(F) Fourth level, a processing-thread-dedicated objects collection, where an object of said collection represents a dedicated area for data holding and data exchange between threads.

[c6] 6. A computer-based method and article of manufacture as per claim 5, further including a method of splitting of a control flow by launching one or more new parallel control flows, wherein said parallel control flow comprises execution of at least one workflow-activity, or sequence of plurality of workflow-activities, before joining with its launching, or other, flow of control, wherein said parallel control flow might optionally launch one or more new parallel control flows, wherein a workflow-activity, that is part of a parallel control flow, generates notification message to synchronizing thread at control flow join-point with other control flow or plurality of control

flows about a potential result with value T of its potential execution, wherein a workflow-activity, that is part of a parallel control flow further splitting one or more times, generates notification messages to synchronizing threads at splitting branches join-points with other control flows about a potential result with value T of its potential execution, wherein synchronizing threads generate notification messages to synchronizing thread responsible for join-point of synchronized control flow with one or more control flows or to synchronizing threads responsible for join-points of said synchronized control flow's splitting branches about potential termination of said synchronized control flow.

[c7] 7. A computer-based method and article of manufacture as per claim 6, further including a method for synchronization of two or more parallel control flows before execution of next in flow-graph workflow-activity according to a synchronization scheme, wherein said synchronization scheme is a conjunction of events signaling completed execution of all workflow-activities represented by sourcing nodes of two or more flow-graph control connectors with destination node being said workflow-activity that will be executed after said synchronization as part of a potential single workflow-request, wherein implementation of said synchronization

method applies to entire multitude of concurrently processed potential workflow-requests.

[c8] 8. A computer-based method and article of manufacture as per claim 7, further including a method of launching alternative control flow routes, wherein said launching optionally takes place where potential execution of a workflow-activity has a result with value N, wherein said launching redirects workflow execution from its normal routes by transferring execution control to one or more workflow-activities with non-synchronized execution, wherein an alternative control-connector launching an alternative control flow route and bypassing one or more workflow-activities with synchronized execution, is coupled with notification-connectors having same source node as said alternative control-connector, wherein each one of said notification-connectors has one of bypassed workflow-activities with synchronized execution as its destination node.

[c9] 9. A computer-based method and article of manufacture as per claim 8, further including construction of hierarchical structure of threads with four levels, wherein in said hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions, wherein each said division is autonomous and self-contained in conducting its tasks, wherein top level

thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of system resources based on its own assessment of application needs and goals, wherein said structure of threads provides capacity for concurrent processing of multitude of requests, limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic, wherein said hierarchical structure of threads contains following levels:

(A) First level, formed by created and activated processing threads, wherein threads forming this level of said hierarchical structure are directly responsible for transactional processing of requests for work and for transactional flow of control between workflow-activities;

(B) Second level, formed by created and activated supervising threads, wherein threads forming this level of said hierarchical structure are responsible for assignment of requests to individual processing threads and supervision of requests" execution;

(C) Third level, formed by all dispatching or synchronizing-dispatching threads being created and activated according to number of workload-dispatching objects in hierarchical tree of class objects and num-

ber of optional workflow-synchronizing objects in hierarchical tree of class objects, wherein dispatching threads being part of Third level are responsible for dispatching of requests to supervising threads, wherein synchronizing-dispatching threads being part of Third level are responsible for synchronization of execution of parallel control flows and for dispatching of requests to supervising threads;
(D) Fourth level, wherein Fourth level is hierarchy's top level and comprises only one thread being responsible for making, executing and supervising decisions about allocation and de-allocation of system resources based on its own assessment, wherein said allocation and de-allocation takes form of modification of First and Second levels of hierarchical structure of threads and their corresponding objects and collections of hierarchical tree of class objects.

[c10] 10. A computer-based method and article of manufacture as per claim 9, further including a method for transactional plugging of software components into workflow-process comprising following steps:
(A) Creation and initialization of processing threads, wherein during its initialization each said processing thread instantiates a non-transactional component object and sends to it an amount of data being con-

stant between processing of individual workflow requests and being necessary to perform execution of relevant portion of workflow request that will potentially be assigned to said processing thread;

(B) Instantiated at step (A) non-transactional component object instantiates its own transactional component-intercepting object, and with GUID (Globally Unique Identifier), sent to it as parameter, instantiates a workflow-activity component-intercepting object;

(C) Supervising thread receives potentially arriving multitudes of single workflow requests and assigns each one of said requests to individual processing thread of its pool;

(D) A processing thread having an assigned at step (C) request for work invokes a method of its non-transactional component object;

(E) An invoked at step (D) non-transactional component object method calls a method of its transactional component-intercepting object, wherein said call of transactional component-intercepting object method constructs transactional component object thereby creating a new transaction and calls a method of said transactional component;

(F) Within context of created at step (E) transaction, said transactional component object method calls a

method of instantiated at step (B) workflow-activity component-intercepting object, wherein said method call instantiates a workflow-activity software component and invokes a method of said component.

[c11] 11. A computer-based method and article of manufacture as per claim 10, further including workload balancing structured at two levels, wherein upper level of said workload balancing comprises multitude of associations between a dispatching thread and multitude of supervising threads and involves dispatching thread balancing workload between its associated supervising threads, wherein lower level of said workload balancing comprises multiple groupings of processing threads in pools associated with a supervising thread per pool and involves supervising threads balancing workload between processing threads of their associated pools.

[c12] 12. A computer-based method and article of manufacture as per claim 11, further including software bottlenecks' prevention and neutralizing, wherein said software bottlenecks' prevention involves encapsulation of a thread pool containing fixed number of processing threads with a supervising thread in a processing-pipe, wherein said software bottlenecks' neutralizing comprises construction of additional processing-pipes and inclusion of constructed additional processing-pipes in

workload balancing process related to workflow-activity where development of bottleneck has been detected.

[c13] 13. A computer-based method and article of manufacture as per claim 12, further including method and apparatus for automatic detection of conditions requiring workflow application scaling up, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of idle threads in processing-pipe's pool reached its critical minimum.

[c14] 14. A computer-based method and article of manufacture as per claim 13, further including method and apparatus for automatic workflow application scaling up, wherein said application scaling up is automatically triggered at a particular workflow-activity to counteract development of a bottleneck at that particular workflow-activity and automatically triggered at all application's workflow-activities for higher application responsiveness when workload increases, wherein said application scaling up involves creation and activation of an additional processing-pipe and inclusion of said additional processing-pipe in workload balancing scheme.

[c15] 15. A computer-based method and article of manufac-

ture as per claim 14, further including method and apparatus for automatic detection of conditions requiring workflow application scaling down, wherein said automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum.

[c16] 16. A computer-based method and article of manufacture as per claim 15, further including method and apparatus for automatic workflow application scaling down, wherein said application scaling down is automatically triggered to counteract a detected inefficiency in use of system memory and CPU time slice allocated to application threads.

[c17] 17. A computer-based method and article of manufacture as per claim 16, further including real-time visualization of quantity, structure, and utilization of threads forming First and Second levels of hierarchical structure of threads and said hierarchical structure's adaptation-enacted modifications of its First and Second levels, wherein said visualization might be used as indicator of workload, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads.

[c18] 18. A computer program product for graphical development of fully executable workflow applications as per claim 1.

[c19] 19. A computer program product that executes as operating system service on a computer running under control of operating system capable of executing multiple execution threads, uses a facility for message queuing and transactional messaging for signaling requests for execution of workflow-activities, uses a facility for distributed transaction coordination, and enacts behavior and capabilities as per claim 4.

[c20] 20. A computer program product for real-time visualization of quantity, structure, and utilization of threads of fully executable workflow applications with adaptive high-performance capacity as per claim 17.